

REMARKS

Claims 1-7 are pending. Claims 1-7 stand rejected. Applicant hereby amends claims 1 and 5.

Rejection under 35 U.S.C. § 103

Claims 1-7 have been rejected under 35 U.S.C. § 103(a) over U.S. Patent 5,991,543 to Amberg et. al. (hereinafter "Amberg") in view of U.S. Patent 5,854,924 to Rickel et. al. (hereinafter "Rickel").

Amberg discloses a method for installing and/or testing software for a build-to-order computer system having a plurality of components includes a plurality of elements. (Abstract) Rickel discloses a static debugging tool for use with a computer and for debugging a binary program file includes an analyzer for causing the computer to statically analyze a representation of a binary program file to detect the presence of program errors and potential errors in the binary program file without executing the binary program file. (Abstract)

Neither Amberg, Rickel, nor the combination of Amberg and Rickel teach or suggest all the claim limitations of claim 1. Applicant has amended claim 1 to further distinguish limitations not taught in Amberg and Rickel.

Amended claim 1 recites, in part, dynamically generating on a simulation computer a file that contains instructions that when executed simulates the process of downloading and the installation of customer ordered software to a target computer.

Amberg teaches a step computer 140 with a sequencing program 204 that sequences software installation and/or testing steps and then writes a series of output files

to step disk 150 for executing the software installation and/or testing steps upon target system 160. (col. 3, line 48 – col. 4, line 17). Applicant fails to see how Amberg teaches a file that contains instructions that when executed simulates the process of downloading and the installation of customer ordered software to a target computer as recited in claim 1 because sequencing steps for software installation and/or testing does not simulate the process of downloading and installation of software. One of the differences between Amberg and claim 1 is that the output file in Amberg executes the software installation and/or testing steps upon the target system. On the other hand, the file in claim 1 when executed simulates the process of downloading and the installation of software to a target computer.

The Office Action recites how Amberg “does not explicitly disclose generating a file on a simulation computer.” The Office Action does recite that Rickel discloses generating a file on a simulation computer. In Rickel, the file is an intermediate, machine independent file that is translated from a binary program file 10 by a decompiler 14 (col. 4, line 59-61). There is no teaching or suggestion in Rickel of a file that simulates the process of downloading and the installation of software to a target computer as recited in claim 1.

Additionally, Applicant fails to see how Rickel teaches a simulation computer. The computer in Rickel is used for static debugging. The analyzer in Rickel statically analyzes a representation of a binary program file to detect the presence of program errors and potential errors in the binary program file without executing the binary program file. (Abstract) Rickel analyzes the intermediate file translated from the binary program file for programming errors such as acceptable ranges for particular values,

function argument and return type, details about what global data functions may read or modify, details about what data pointed at by arguments may be read or modified, and details on how to interpret arguments to functions that might take different numbers and types of arrangements. (col. 4, lines 5-10). The static analysis in Rickel for programming errors does not teach or suggest a simulation computer because no simulation of the execution of the intermediate file occurred in Rickel. Thus, Rickel does not teach or suggest dynamically generating on a simulation computer a file that contains instructions that when executed simulates the process of downloading and the installation of customer ordered software to a target computer as recited in claim 1.

Claim 1 also recites simulating the execution of said dynamically generated file in accordance with a set of evaluation rules such that the outcome of the execution of said file is determined. The Office Action does recite that Amberg does not explicitly disclose simulating the execution of said dynamically generated file in accordance with a set of evaluation rules. The Office Action cites col. 1, line 55 – col. 2, line 9 in Rickel and recites that “[t]he static debugging tool includes an analyzer for causing the computer to statically analyze (i.e. simulate) a representation of a . . . file to detect the presence . . . of program errors . . . without executing the . . . file.” Applicant traverses this assertion that the static analysis in Rickel is equivalent to simulation. As discussed above, Rickel analyzes the immediate file translated from the binary program file to identify programming errors. Rickel clearly states that no execution of the intermediate file occurs (Summary). Also, identifying programming errors in an intermediate file does not suggest simulating the execution of a dynamically generated file as recited in claim 1. Thus, absent a teaching of simulation in Rickel, claim 1 is allowable.

The Office Action recites that the modification to combine Rickel and Amberg would have been obvious because one of ordinary skill in the art would have would “want to save time and testing costs by generating and testing the file on the simulation computer.” Additionally, the Office Action recites that one of ordinary skill in the art would have wanted to analyze the outcome of the simulation of the execution of said file in order to find and correct possible syntax and flow errors in order to produce a defect-free file without the risk of expense of actually executing the file to identify the errors. The Office Action also recites that Rickel is in an analogous environment to Amberg. Applicant respectfully traverses these assertions. First, Rickel statically analyzes binary program files for programming errors. This kind of static debugging occurs at the program code level. The installation and/or testing of the software in Amberg occurs at the application level. Second, the kind of debugging in Rickel occurs in the software development phase as recited in col. 1, lines 4-5. In Amberg, the installation and/or testing of the software is for a build-to-order computer system. At this point, the software is already debugged for programming errors and is being prepared to be loaded onto a build-to-order computer system. Thus, one skilled in the art would not have combined Rickel and Amberg because the debugging of Rickel occurs at the program code level, while the installation and testing of software occurs at the application level. Moreover, the debugging tool in Rickel occurs in software development, which is a separate process and occurs well before manufacturing of a build-to-order computer system in Amberg.

Thus, claim 1 is allowable over Amberg and Rickel for at least the above stated reasons.

Claims 2-4 are dependent either directly or indirectly on claim 1 and are allowable for at least the same reasons as claim 1.

Claim 5 includes similar limitations of “a simulation computer” and “simulates the process of downloading and the installation of customer ordered software onto the target computer” as in claim 1 and is allowable for at the above stated reasons for claim 1.

Claims 6-7 are dependent either directly or indirectly on claim 5 and are allowable for at least the same reasons as claim 5.

CONCLUSION

Based on the foregoing amendments and the remarks, Applicant believes that the rejections in the Final Office Action are fully overcome, and that the Application is in condition for allowance. If the Examiner has questions regarding the case, the Examiner is invited to contact Applicant's undersigned representative at the number given below.

Respectfully submitted,
Steven E. Baker

Date: 6/8/05

By: Eugene G. Kim
Eugene G. Kim, Reg. No. 46,267
Carr & Ferrell LLP
2200 Geng Road
Palo Alto, CA 94303
Phone: (650) 812-3400
Fax: (650) 812-3444